

## Assignment 6: CH5350

Abhijeet Mavi

### MLE and Least Squares Estimate

We are given  $x[1]$  and  $x[2]$  here. So, the given equation is.

$$x[k] = -d_1 x[k-1] + e[k]$$

So, we have to evaluate  $d_1$  and  $\sigma_e^2$  here. In order to do so, we will assume that the series is zero-mean stationary, we will get

$$E(x[1]) = 0$$

$$\sigma_{x[1]} = \frac{\sigma_e^2}{1 - d_1^2}$$

Also,

$$E(x[2]|x[1]) = -d_1 x[1]$$

$$\sigma_{x[2]|x[1]}^2 = \sigma_e^2$$

Now for likelihood function, we will write the joint distribution of the following as follows:

$$\begin{aligned} f(x^2; \theta) &= f(x[1]; \theta) * f(x[2]|x[1]; \theta) \\ &= \left( \frac{\sqrt{1 - d_1^2}}{\sqrt{2\pi\sigma_e^2}} \exp\left(-0.5 \frac{x^2[1](1 - d_1^2)}{\sigma_e^2}\right) \right) * \left( \frac{1}{\sqrt{2\pi\sigma_e^2}} \exp\left(-0.5 \frac{(x[2] + d_1 x[1])^2}{\sigma_e^2}\right) \right) \\ f(x^2; \theta) &= \frac{1}{2\pi\sigma_e^2} \exp\left(-0.5 \frac{x^2[2] + 2d_1 x[1]x[2] + x^2[1]}{\sigma_e^2}\right) \end{aligned}$$

We take the log-likelihood as follows,

$$L = c + \frac{1}{2} \ln(1 - d_1^2) - 2 \ln(\sigma) - \left( \frac{1}{2} \frac{x^2[2] + 2d_1 x[1]x[2] + x^2[1]}{\sigma_e^2} \right)$$

Now, for maximum likelihood of  $\theta = [\sigma_e^2 \ d_1]$ , we have  $\frac{\partial L}{\partial \sigma_e^2}$  and  $\frac{\partial L}{\partial d_1}$ . Then, I solved the 2-equation, 2-variable system, we get:

$$\begin{aligned} \hat{d}_1 &= \frac{-2x[1]x[2]}{x^2[1] + x^2[2]} \\ \hat{\sigma}_e^2 &= \frac{(x^2[1] - x^2[2])^2}{2(x^2[1] + x^2[2])} \end{aligned}$$

For the Least-square estimates,

$$x[2] = -d_1 x[1] + e[2]$$

Hence, the least squares solution for this:

$$\hat{d}_1 = -\frac{x[2]}{x[1]}$$

Hence, we get

$$E((x[2] - \hat{x}[2])^2) = 0$$

Hence, there is no noise in variance, as  $\hat{\sigma}_e^2 = 0$ .

As, the variance of LS estimate is much lower than the MLE estimator, I believe that **LS is a better estimator in this case.**

## Hannan-Rissanen Algorithm

```
arma_hn=function(series,ar_order,ma_order){  
  len_ser=length(series)  
  max_order=5  
  ar5mod=arima(series,c(5,0,0))
```

I formulated an AR(5) model to first find the coefficients of  $e[k]$ . This formulated noise would then serve as regressors for moving average part of the ARMA models. Then, the  $z\_mat$  formulated in the code snippet below served as regressor to our overall problem statement, and we applied the least squares solution according to Projection theorem as follows.

$$\hat{\phi} = (Z^T Z)^{-1} Z^T Y$$

where  $\hat{\phi}$  served as the estimated parameter set

```
res_ek=ar5mod$residuals  
res_ek=res_ek[which(is.na(res_ek)==F)]  
ser_new=series[max(ar_order,ma_order):len_ser]  
z_mat=matrix(NA,nrow=(len_ser-(max(ma_order,ar_order))-  
1),ncol=(ar_order+ma_order))  
for(i in 1:(ar_order)){
```

```

    z_mat[,i]=series[(max(ar_order,ma_order)-i+2):(len_ser-i)]
}

for(i in 1:(ma_order)){
    z_mat[, (i+ar_order)]=res_ek[(max(ar_order,ma_order)-i+2):(len_ser-i)]
}

par_set=((qr.solve(t(z_mat) %*% z_mat)) %*% t(z_mat)) %*%
series[(max(ar_order,ma_order)+2):len_ser]

return(par_set)
}

ak=arima.sim(n = 100000, list( ar=c(0.4),ma = c(0.7, 0.12)),sd = sqrt(1))
ak1=arima.sim(n = 100000, list( ma = c(1, 0.21)),sd = sqrt(1))
arma_hn(ak1,0,2)

arma_hn(ak,1,2)

```

I then supplied MA(2) process with  $c_1 = 1$ ,  $c_2 = 0.21$  and ARMA(1,2) process with  $d_1 = 0.4$ ,  $c_1 = 0.7$ ,  $c_2 = 0.12$ .

For MA(2), I got  $c_1$  as **0.99** and  $c_2$  as **0.20**. For ARMA(1,2), we got  $d_1$  as **0.41**,  $c_1$  as **0.69** and  $c_2$  as **0.11**. These values are quite close to the real values.

```

library(tseries)

ak=arima.sim(n = 100000, list( ar=c(0.4),ma = c(0.7, 0.12)),sd = sqrt(1))
ak1=arima.sim(n = 100000, list( ma = c(1, 0.21)),sd = sqrt(1))

arma12mod=arma(ak,c(1,2))
arma12mod$coef

##          ar1          ma1          ma2  intercept
## 0.38867869 0.71293418 0.13396257 0.00400538

ma2mod=arma(ak1,c(0,2))
ma2mod$coef

##          ma1          ma2  intercept
## 0.99903690 0.20700842 -0.01039001

```

Again, the values are closer to the **arma()** routine as shown.

## Spectral Densities

We see that the given model is

$$x[k] = e[k - 2] + 2e[k - 1] + 4e[k]$$

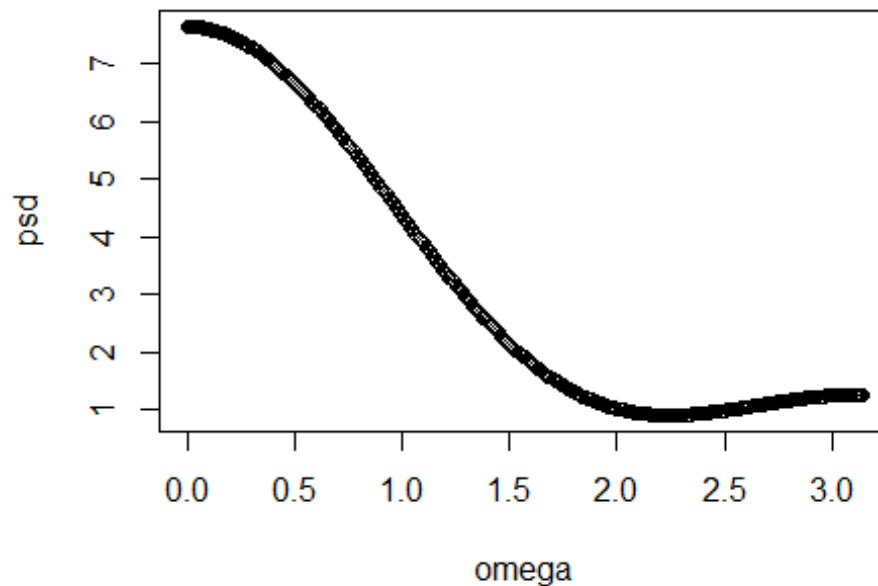
We can find  $\gamma(\omega)$  using the function

$$\gamma(\omega) = \frac{\sum_{l=-\infty}^{l=\infty} \sigma[l] e^{-j\omega l}}{2\pi}$$

We find that,

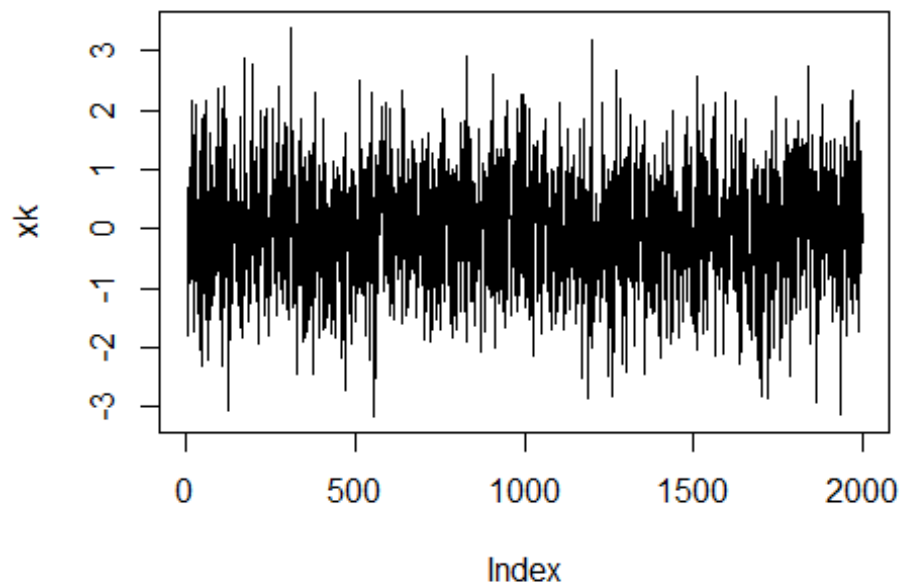
$$\gamma(\omega) = \frac{\sigma_e^2}{2\pi} [20 + 20\cos(\omega) + 8\cos(2\omega)]$$

```
psd={}
omega=seq(0,3.14,by=0.01)
for (i in 1:length(omega)){
  psd[i]=(20+(20*cos(omega[i]))+(8*cos(2*omega[i])))/6.28
}
plot(omega,psd)
lines(omega,psd)
```



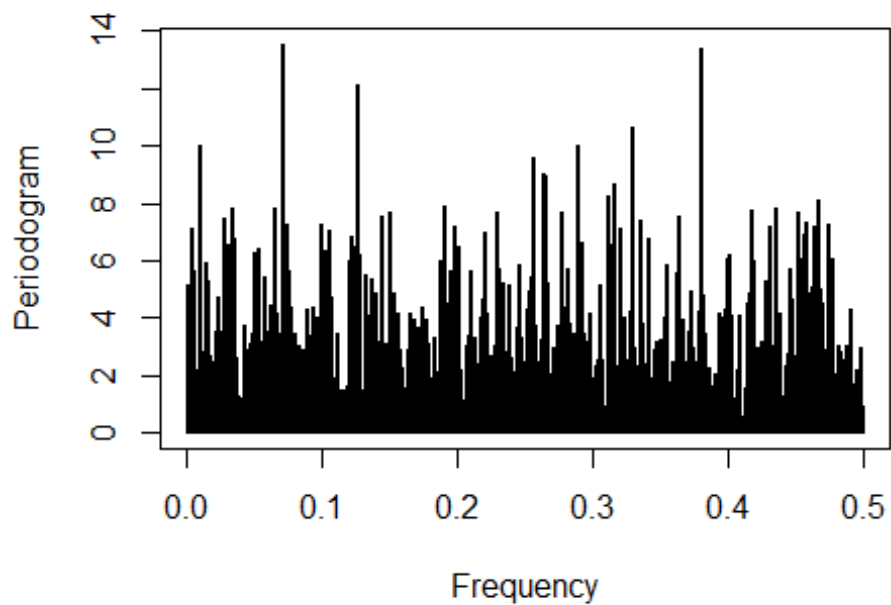
Now, we will generate 2000 samples of the said dataset as follows

```
ek=rnorm(10000)
xk=ek[4001:6000]
plot(xk, type='l')
```



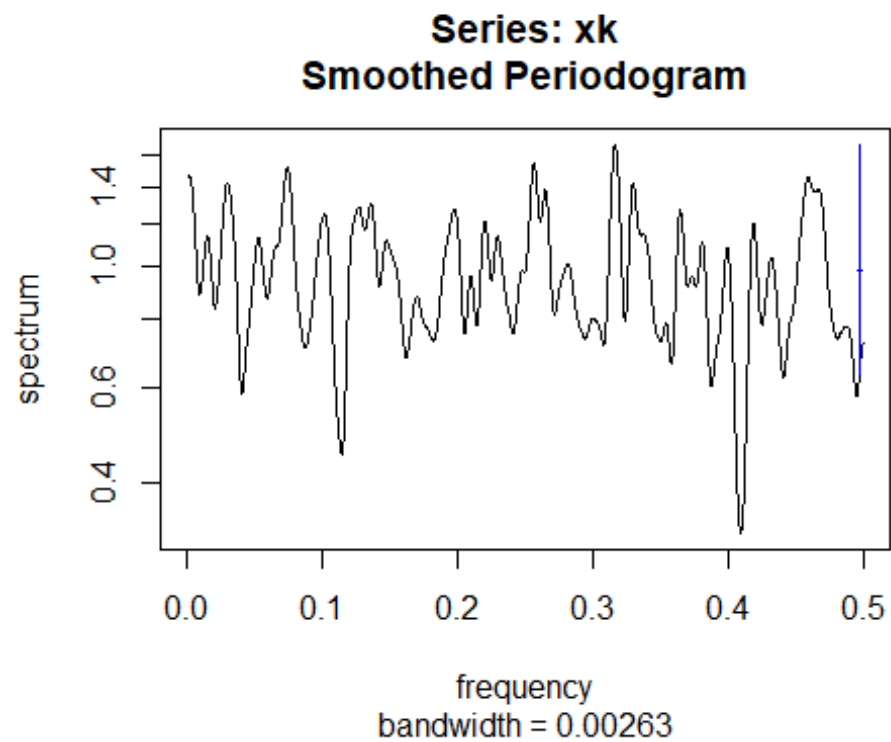
We then apply periodogram to it as follows:

```
library(TSA)
##
## Attaching package: 'TSA'
##
## The following objects are masked from 'package:stats':
##
##   acf, arima
##
## The following object is masked from 'package:utils':
##
##   tar
periodogram(xk)
```



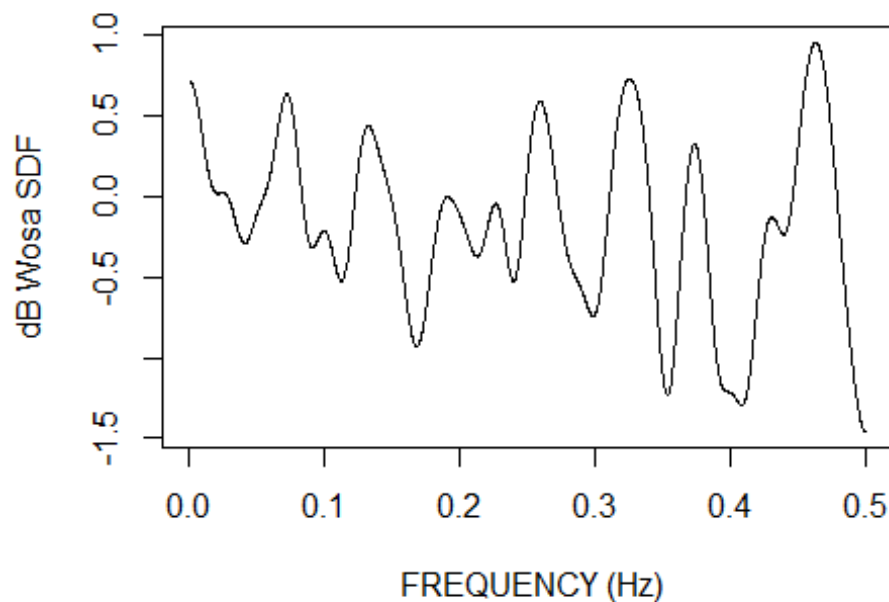
*# For Daniell's method*

```
spec.pgram(xk , spans=c( 9 , 9 , 9 , 9 , 9 ))
```



The Welch's periodogram is plotted using **SDF()** function as follows:

```
library(sapa)
plot(SDF(xk ,method="wosa" , blocksize= 70))
```



We will now try to fit an MA(2) model for our process, and then try to calculate the PSD of it using parameters  $c_1$  and  $c_2$ . We know that  $\sigma[0] = 1 + c_1^2 + c_2^2$ ,  $\sigma[1] = c_1 + c_1c_2$  and  $\sigma[2] = c_2$  for an MA(2) process.

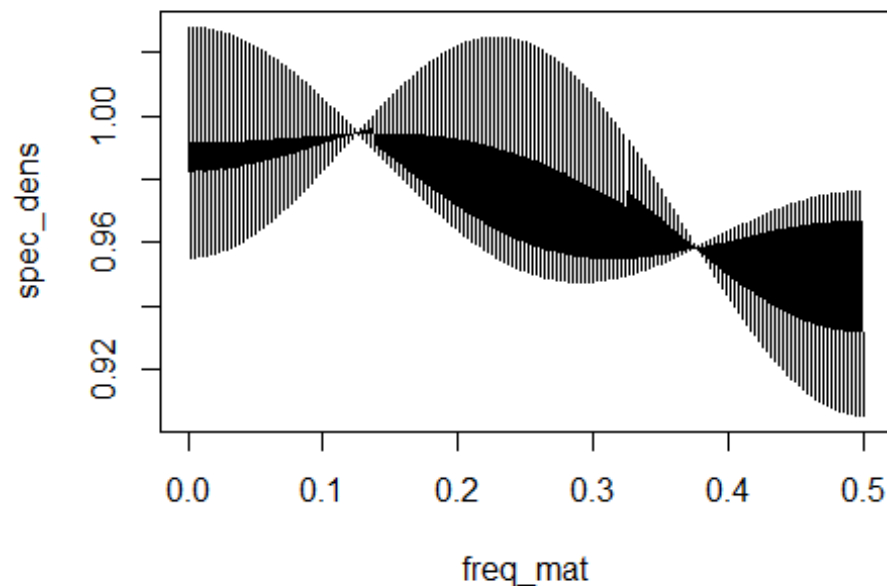
```
ma2mod=arima(xk,c(0,0,2))
c1=ma2mod$coef[1]
c2=ma2mod$coef

err_sig=ma2mod$sigma2

freq_mat=seq(0,0.5,by=0.001)

spec_dens=
(err_sig)*((1+(c1^2)+(c2^2))+2*(c1+(c1*c2))*cos(2*pi*freq_mat)+2*c2*cos(2*2*pi*freq_mat))

plot(freq_mat,spec_dens,type='l')
```



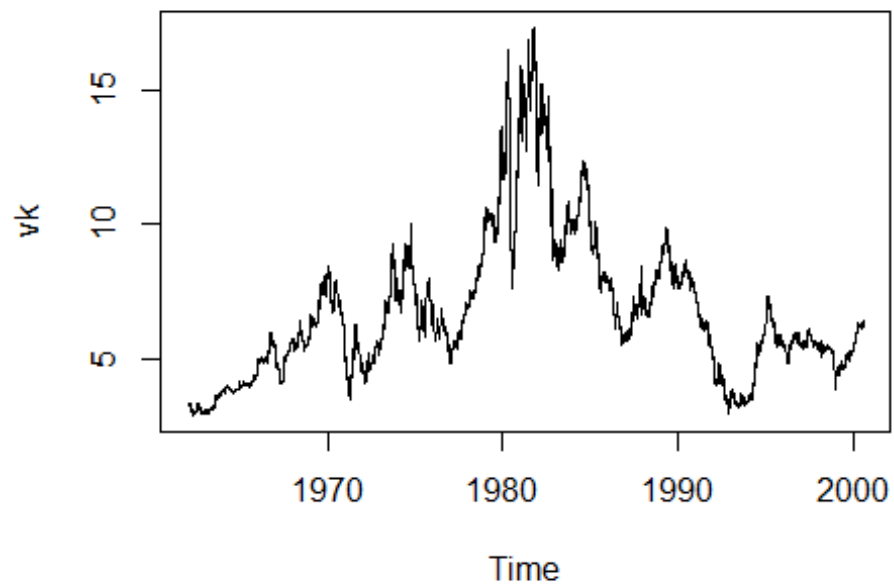
It is quite clear from the whole process that parametric method gives the best form of PSD, even though the graph does not accurately describe our process but it gives the smoothest approximation of all as compared to non-parametric ones. Note that, for the original process I have plotted  $\omega$  and for subsequent processes, I plotted frequencies.

### Fitting a time-series model: tcm1yd

```
library(tseries)
library(TSA)
data(tcmd)
vk=tcm1yd

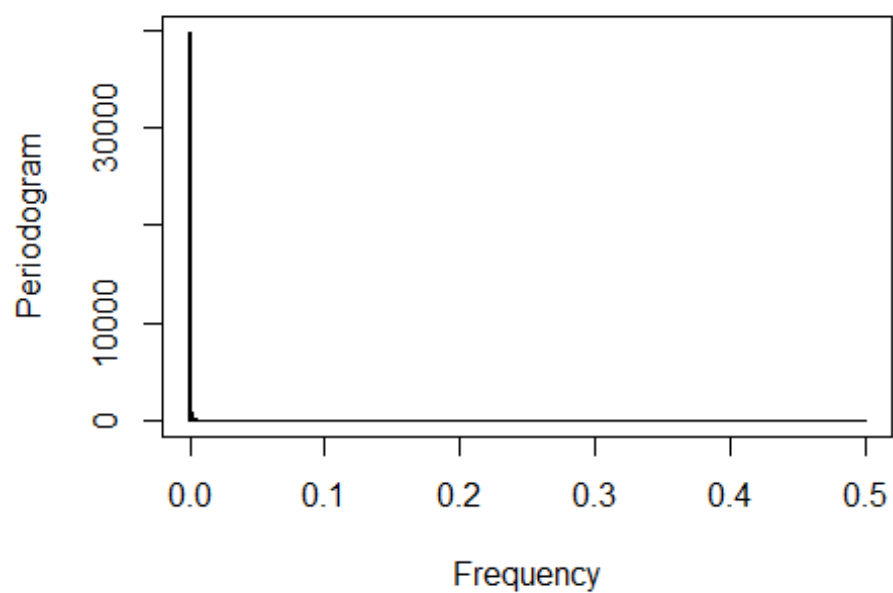
plot(vk)
```



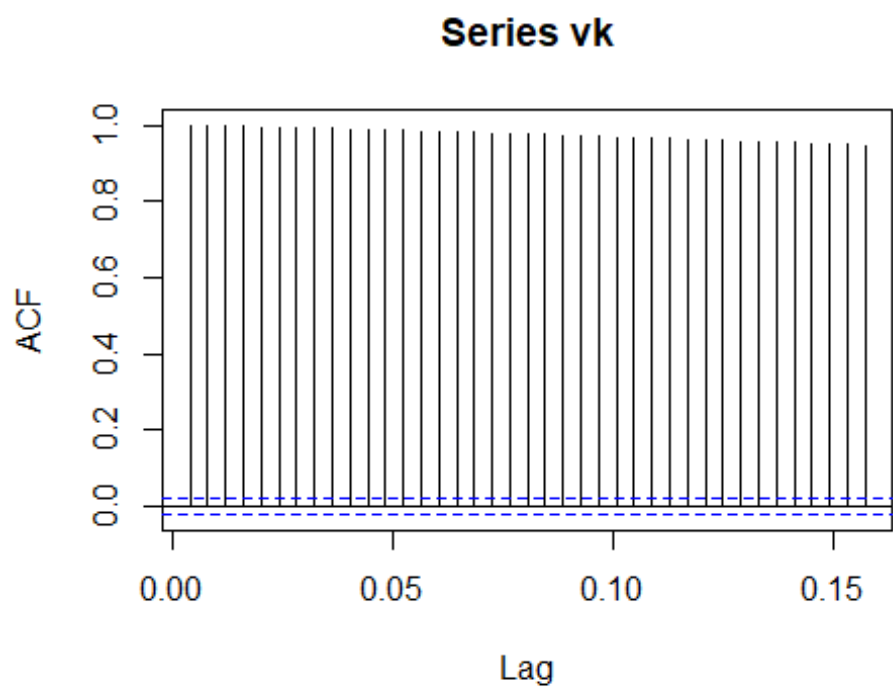


We first load the tcmd model and, and then take tcm1yd from it. We then plot it as shown above.

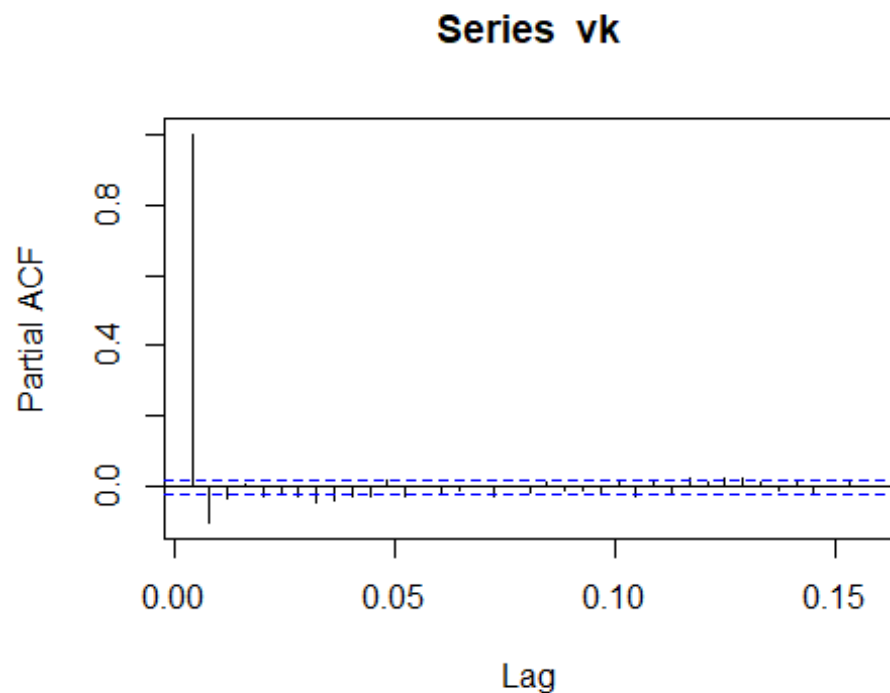
```
periodogram(vk)
```



`acf(vk)`



`pacf(vk)`



The periodogram, acf, and pacf clearly shows that there are trends in the series. Particularly differencing trends. This is because the decay in acf is too slow and the periodogram shows high power at lower frequency, a characteristic of differencing series.

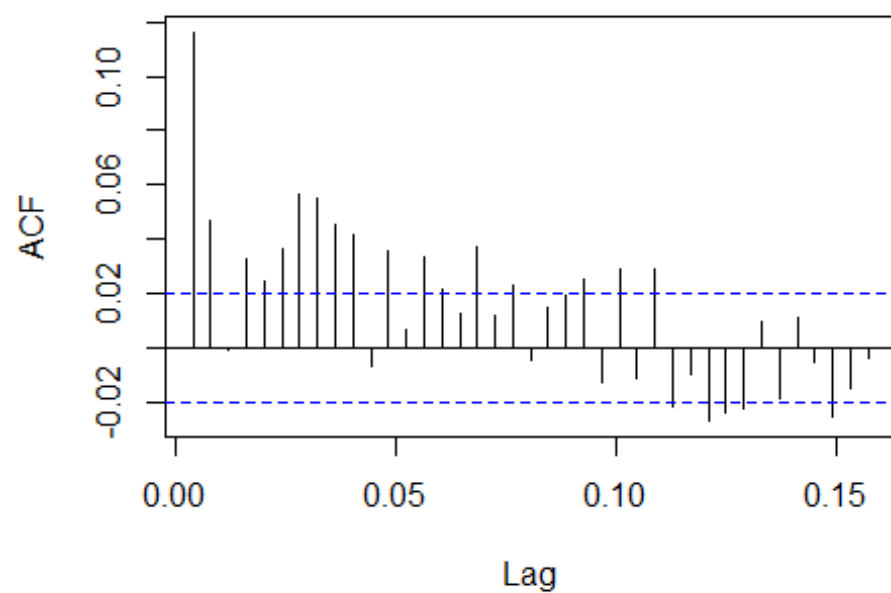
```
diff_vk= diff(vk)

adf.test(diff_vk)

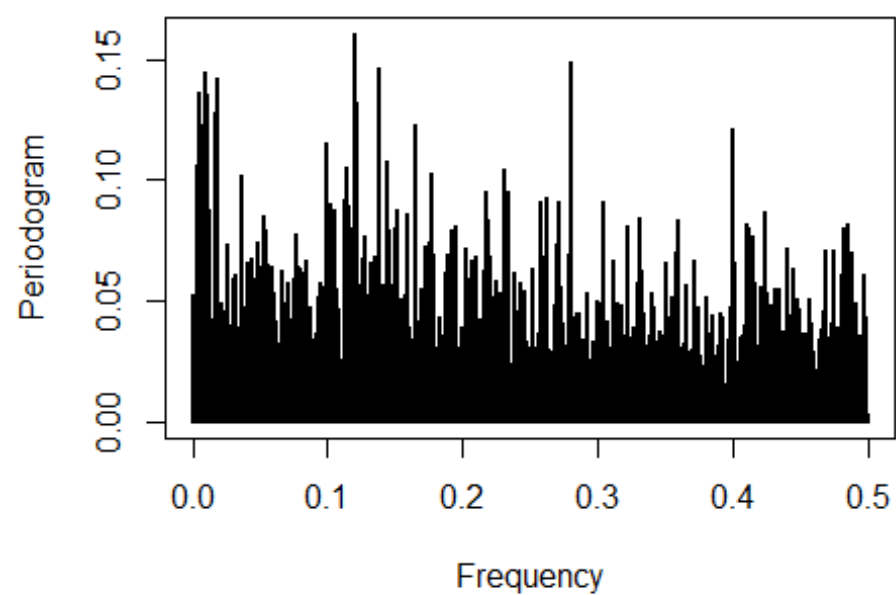
## Warning in adf.test(diff_vk): p-value smaller than printed p-value
##
## Augmented Dickey-Fuller Test
##
## data: diff_vk
## Dickey-Fuller = -17.642, Lag order = 21, p-value = 0.01
## alternative hypothesis: stationary

acf(diff_vk)
```

**Series diff\_vk**



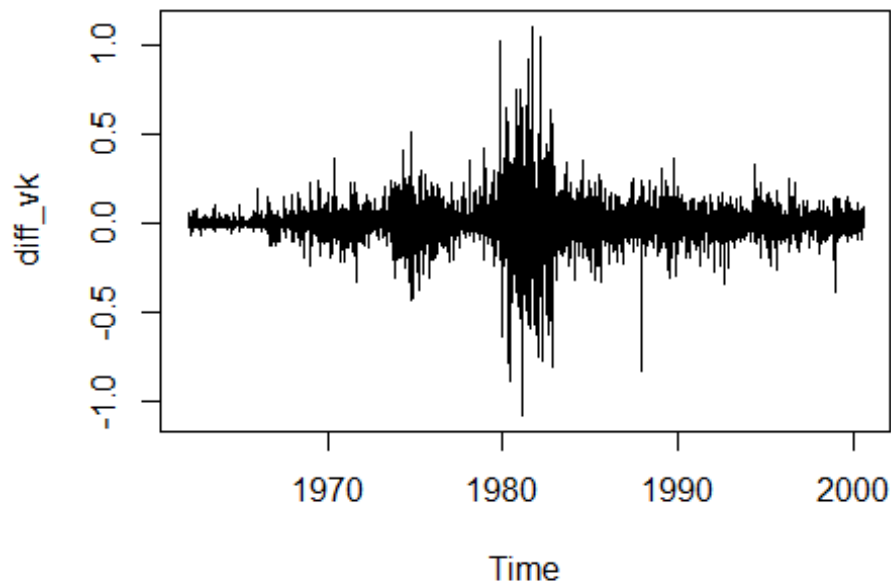
```
periodogram(diff_vk)
```



I then differenced the series once, and then applied the Augmented Dickey-Fuller test using **adf.test()**. It clearly showed that null hypothesis can be rejected of the series requiring any more differencing.

The periodogram further reaffirmed my hypotheses as shown.

```
plot(diff_vk)
```



```
arma33mod=arima(diff_vk, order=c(3,0,3))
```

```
arma33mod
```

```
##
```

```
## Call:
```

```
## arima(x = diff_vk, order = c(3, 0, 3))
```

```
##
```

```
## Coefficients:
```

```
##          ar1      ar2      ar3      ma1      ma2      ma3  intercept
```

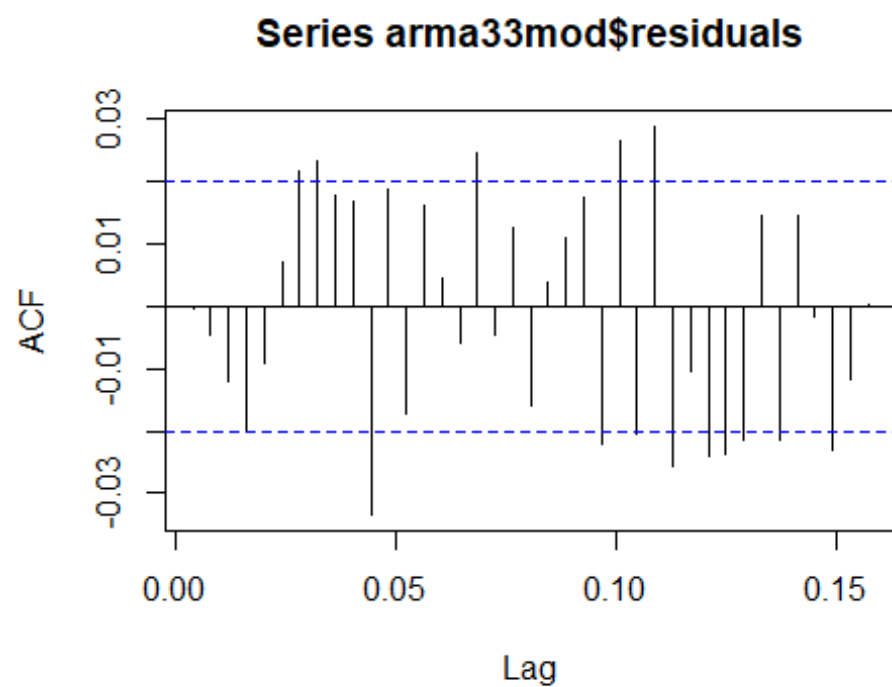
```
##          0.2772  0.2861  0.2988  -0.1692  -0.2708  -0.3410         0.0003
```

```
## s.e.    0.1891  0.1600  0.1007   0.1864   0.1437   0.0947         0.0015
```

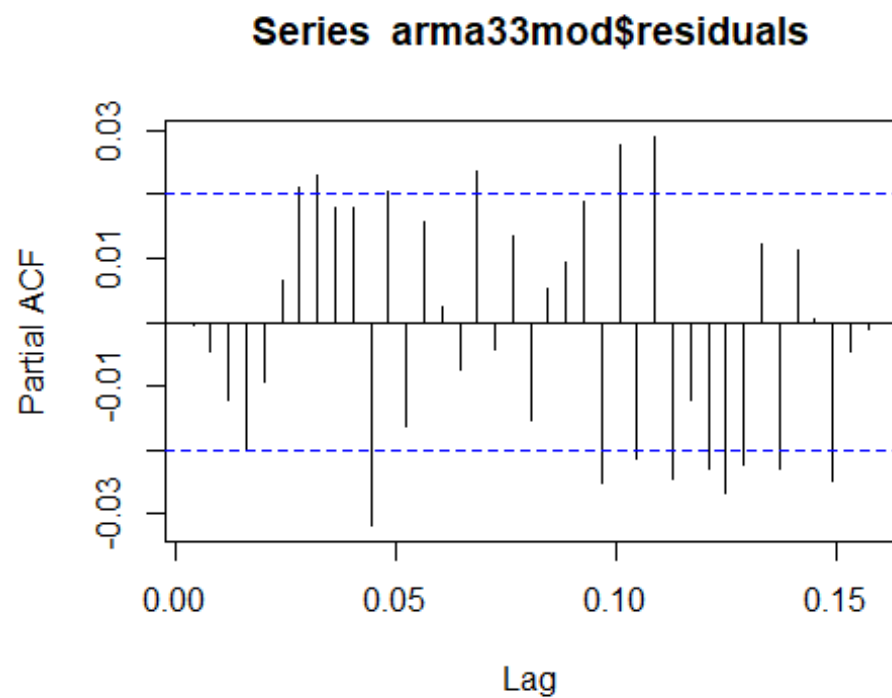
```
##
```

```
## sigma^2 estimated as 0.00904:  log likelihood = 8942.38,  aic = -17870.76
```

```
acf(arma33mod$residuals)
```



```
pacf(arma33mod$residuals)
```



I then tried to fit various ARMA models to the series and was able to properly fit the ARMA(3,3) to the differenced series. Even though it showed a bit of underfitting discrepancies, it fit our model the best in all the possible models of 10th order.

Hence, our final series looks like as follows:

$$\begin{aligned} & \nabla v[k] \\ = & 0.2772v[k-1] + 0.2861v[k-2] + 0.2988v[k-3] - 0.1692e[k-1] - 0.2708e[k-2] \\ & - 0.3410e[k-3] + e[k] \end{aligned}$$

The estimated  $\sigma_e^2$  is **0.00904**.

## Predictions using Projection Theorem

For the given series, we have

$$x[k] = e[k] - e[k-1]$$

For any process, the best predictor is the linear sum of its past realizations,

$$\hat{x}[k|k-1] = \sum_{i=0}^{k-1} \phi[i]x[k-i]$$

We see that  $\sigma[0] = 2\sigma_e^2$  and  $\sigma[1] = -\sigma_e^2$ . Now, using Yule-Walker's equations we construct the following matrix.

$$\sigma_e^2 \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \dots & 0 \\ 0 & -1 & 2 & \dots & 0 \\ \dots & \dots & \dots & \dots & 2 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \dots \\ \phi_k \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ \dots \\ 0 \end{pmatrix}$$

I solved the given matrix system to find that,

$$\begin{pmatrix} \phi_1 \\ \phi_2 \\ \dots \\ \phi_k \end{pmatrix} = \begin{pmatrix} -\frac{k}{k+1} \\ \frac{k-1}{k+1} \\ -\frac{k-2}{k+1} \\ \dots \\ 1 \\ -\frac{1}{k+1} \end{pmatrix}$$

And hence,

$$\hat{x}[k|k-1] = -\sum_{i=1}^k \frac{k-i+1}{k+1} x[k-i]$$

Now, in order to calculate MSE for the above derived BLP estimate, we have

$$MSE = E((x[k] - \hat{x}[k|k-1])^2)$$

$$MSE = E((x[k] + \sum_{i=1}^k \frac{k-i+1}{k+1} x[k-i])^2)$$

$$MSE = 2\sigma_e^2 - 2\frac{k}{k+1}\sigma_e^2 + E(\sum_{i=1}^k (\frac{k-i+1}{k+1})^2 x[k-i]^2) + E(\sum_{i=1}^k \sum_{j=1}^k \frac{k-i+1}{k+1} \frac{k-i}{k+1} x[k-i]x[k-i-j])$$

We will only see j for j=1, since all the other covariances are 0. Expanding, the expectations terms, we get

$$MSE = 2\sigma_e^2 - 2\frac{k}{k+1}\sigma_e^2 + 2E(\sum_{i=1}^k (\frac{k-i+1}{k+1})^2 x[k-i]^2) + 2E(\sum_{i=1}^k \frac{k-i+1}{k+1} \frac{k-i}{k+1} x[k-i]x[k-i-1])$$

The third and fourth terms are happening when l=0 and l=1.

Further, simplifying, we would get,

$$\frac{2\sigma_e^2}{k+1} + 2\sigma_e^2 \sum_{i=1}^k (\frac{k-i+1}{(k+1)^2}) = \frac{2\sigma_e^2}{k+1} (1 + \frac{k}{2}) = \frac{k+2}{k+1} \sigma_e^2$$

Hence, proved that MSE is  $\frac{k+2}{k+1} \sigma_e^2$ .